EXHIBIT A

Optimal Selection with Stability of Power Flow in a Hybrid Electric Vehicle

This invention is a computer control algorithm for hybrid electric vehicles. Given the present operating conditions of the propulsion system, the control algorithm searches through the feasible solution space of power flows to obtain the one that best satisfies a prescribed objective cost (penalty) function. Power flows refer to the magnitude and direction of power flow between the engine, electric motor/generators, battery pack and vehicle. Thus, a determination of the optimal power flow allows the trade-off of fuel consumption to battery energy consumption to be made.

Quiescent conditions in the vehicle and propulsion system are detected, and changes in the propulsion system operating point are further penalized when conditions are quiescent.

The control algorithm achieves optimal trade-offs between propulsion system and vehicle-level goals while still operating within the system's constraints. For example, fuel economy and emissions are optimized subject to cost (penalty) functions for energy storage usage and constraints based on energy storage system charge state.

The generalized steps are as follows. The details behind each step are explained thereafter:

1. Determine the feasible solution space in the independent variable.
2. Select a value for the independent variable from within the valid solution space.
3. Compute the dependent variable at this point.
4. Determine whether the dependent variable is within its constraints. If outside, skip to step 7.
5. Compute the cost function value at this operating point.
6. Compare the cost function value at this operating point to that at other operating points.
7. Eliminate the solution space in the independent variable beyond the point which has the higher cost function (or infeasible dependent variable).
8. Repeat Steps 2-7 to desired convergence level (a fixed number of iterations or small cost function difference).
9. If quiescent conditions are detected, add an additional penalty to the cost function value at this operating point.
10. Recall from memory the independent variable with least cost from the last control loop.
11. Compute the dependent variable at this operating point.
12. Determine whether the dependent variable is within its constraints. If outside, skip to step 14.
13. Compute the cost function value at this operating point.
14. Eliminate the independent variable with higher cost (or infeasible dependent variable).
15. Store in memory the selected independent variable.

The remaining value of the independent variable becomes the torque command to the engine.

The following sections further explain these steps as they pertain to the optimal selection with stability of power flow in a hybrid electric vehicle.

1. Determining the Feasible Solution Space

The system equations are derived and placed into the following form for control:

$$\begin{bmatrix} Ta \\ Tb \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} \\ K_{21} & K_{22} & K_{23} & K_{24} \end{bmatrix} \begin{bmatrix} Ti \\ To \\ Ni\_dot \\ No\_dot \end{bmatrix}$$

where Ta = Unit A Torque
Tb = Unit B Torque
Ti = Input Torque
To = Output Torque

EXHIBIT A

Ni_dot = Input Acceleration
No_dot = Output Acceleration
Kn = System constants including reflected gear ratios and inertias

A novel method is used to determine the bounds of feasible output torque (ROI to be submitted). The driver's requested output torque, To, is subjected to these constraints. The input acceleration, Ni_dot, is a desired quantity. The output acceleration, No_dot, is a measured quantity. Only the input torque, Ti, remains as an unknown independent variable on the right half of the equation.

A novel method is used to determine the range of feasible input torque, Ti, from which to select (ROI to be submitted). The high and low boundary values of Ti are designated *Ti_bound_lo* and *Ti_bound_hi.*

2. Select a Value for the Independent Variable
(Select input torque, *Ti*)

From the range of feasible input torque, Ti, select a value Ti that will later be evaluated.

One method to select the Ti which is not claimed under this invention is the golden section ratio (0.61803...). Using this ratio to select the value of the independent variable on the range of feasible values can be shown to converge to a solution most quickly. Other approaches could be used to select the independent variable.

3. Compute the Dependent Variable at this Point
(Compute battery power, *Pbat*)

The key dependent variable of concern is the battery power, Pbat. It is computed by first computing unit A torque and unit B torque at the independent variable (Ti) and then estimating the machine power loss terms to obtain:

$$Pbat = Ta \cdot Na + Pa\_loss + Tb \cdot Nb + Pb\_loss$$

where Na = Unit A Speed
Nb = Unit B Speed
Pa_loss = Unit A electrical power loss
Pb_loss = Unit B electrical power loss

This value of the dependent variable represents the battery power that *would* occur if the given value of the independent variable, Ti, were selected. Additional terms can be added to this equation to account for additional loads on the DC bus such as an auxiliary power converter, Pdc_load.

4. Determine whether the Dependent Variable is Within its Constraints
(whether battery power, *Pbat*, is within the limits *Pbat_min* and *Pbat_max*)

In this algorithm step, the dependent variable, *Pbat*, is compared to its constraint or boundary limits, *Pbat_min* and *Pbat_max*. These limits are obtained using a novel method (ROI to be submitted).

If *Pbat* is outside a limit value, then the corresponding value of the independent variable shall become a new boundary value to the feasible solution space. Skip ahead to step 7.

5. Compute the Cost Function Value at this Operating Point
(Compute Cost at this Input Torque, Ti)

The cost function measures the relative expense of operating the system at one operating point compared to another operating point. The cost function includes two categories: 1) estimates of component power loss

EXHIBIT A

and 2) subjective costs. Each can be calculated or derived from the independent variable (input torque, Ti). The power loss terms include the following:

$$Ploss\_total = Ploss\_engine + Ploss\_motorA + Ploss\_motorB + Ploss\_battery$$

Ploss_engine is determined by look-up table of empirical data. The loss function data is computed by subtracting the actual engine output power from the amount of fuel power required to deliver that output power assuming the engine were performing at its best efficiency.

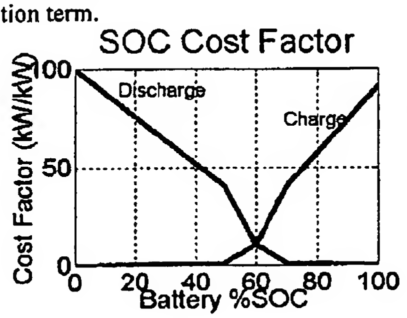
$$P_{loss\ engine} = \eta_{MAX}\ LHV\ (kJ/g)\ Q_{FUEL}\ (g/s) - P_{OUT}$$

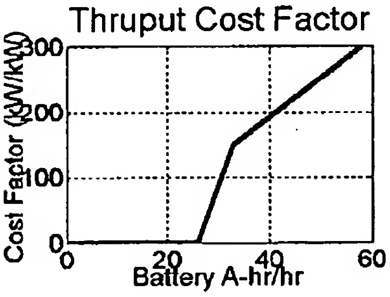
Where η_{MAX} is the engine's maximum efficiency (e.g. 35%), LHV is the fuel's lower heating value (a measure of its energy content per unit mass), Q_{FUEL} is the fuel flow rate at that value of torque and speed, and Pout is the engine's actual mechanical output power at that point.

Ploss_motorA and Ploss_motorB represent the losses in the motor/generator units as well as their controlling power electronics. They are determined from empirical data by the difference in mechanical power (Torque x Speed) and electrical power (Voltage x Current).

Ploss_battery is determined by equations for the battery's internal resistance (I^2R).

Subjective costs are penalties that cannot be derived from physics in terms of units of power loss, but rather represent some other form of penalty against operating the system at that point. These penalties are subjectively scaled with units of power loss so they can be compared equivalently with more physics based losses. A separate ROI describes these subjective losses. The first cost function terms penalize charging at high State-of-Charge and penalize discharging at low State-of-Charge. A second cost function term captures the effect of battery age. If battery age is measured in terms of average battery current, then a penalty may be placed on the operating point that increases with higher battery current. The figures below illustrate Cost Factors for these terms. The product of Cost Factor and Battery power yields the cost function term.





The total subjective cost is then the following summation:

$$P\cos t\_subjective\_total = P\cos t\_SOC + P\cos t\_thruput$$

The total cost is then the following summation:

$$P\cos t\_total = P\cos t\_subjective\_total + Ploss\_total$$

Alternative ways for representing the power loss include look-up tables of empirical data or equations with other unique variables (e.g. temperature). For example the Ploss_battery term could be modified to include internal resistance as a function of temperature. Other terms reflecting power loss could also be included.

EXHIBIT A

Other subjective costs could also be penalized provided they could be derived from the independent variable.

6. Compare the Cost Function Value at this Operating Point to that at other Operating Points (Compare the Cost at this Input Torque, Ti to that at other Input Torques, Ti.)

Compare the Cost at this Input Torque, Ti to that at other Input Torques, Ti. If no prior cost function has been computed, then return to step 2 and select the second operating point (input torque, Ti).

7. Eliminate the Solution Space in the Independent Variable beyond the point which has the higher cost function value (or infeasible dependent variable)

If the value of the cost function at point 1 (designated *Ti_1*) is less than the value of the cost function at point 2 (designated *Ti_2*) (and assuming *Ti_1* < *Ti_2*), then set the new boundary, *Ti_bound_hi* equal to *Ti_2*.

If the value of the cost function at point 2 (designated Ti_2) is greater than the value of the cost function at point 2 (designated Ti_2) (and assuming *Ti_1* < *Ti_2*), then set the new boundary, *Ti_bound_lo* equal to *Ti_1*.

8. Repeat Steps 2-7 to desired convergence level (a fixed number of iterations or small cost function difference.

The input torque, Ti, on this control loop with the minimum cost is designated, *Ti_min_cost*. The value of the cost function here is *Pcost_total_minimum*.

Note: Steps 1-8 can stand and function on their own. The command to the engine becomes, Ti_min_cost. Steps 9-15 add additional robustness to ill-conditioned cost functions (e.g. small changes in cost over wide ranges of torque or more than one local minimum to the cost function)

9. If quiescent conditions are detected, add an additional penalty to the cost function value at this operating point.

Low accelerator pedal position and vehicle speed information are used to determine if conditions are quiescent. When conditions are quiescent, even small amounts of instability or "hunting" in the engine torque command can be observed by the driver and passengers and should be avoided.

```
IF Quiescent_Conditions == FALSE
THEN
        ** Determine if Quiescent conditions exist now
        IF Accelerator_Pedal_Position < Cal_Lo_Accel_Pedal_Threshold
                AND Vehicle_Speed < Cal_Lo_Vehicle_Speed_Threshold
        THEN
                Quiescent_Conditions = TRUE
                Initialize Quiescent_Conditions_Timer to cal_Hold_Engine_Torque_Steady_Time
        ENDIF
ELSE
        ** Determine if Quiescent timer has expired. Periodically (e.g. 30 seconds)
        ** the penalty is removed so the a new optimal value can be found. Undetectable by the driver.
        Decrement Quiescent_Conditions_Timer
        IF Quiescent_Conditions_Timer == 0
        THEN
                Quiescent_Conditions = FALSE
        ENDIF
ENDIF

IF Quiescent_Conditions == TRUE
```

EXHIBIT A

```
THEN
        ** The cost function at the Ti computed for this loop is penalized in order to bias operation toward
        ** the engine torque selected on the last loop.
        Pcost_total_this_loop = Pcost_total_minimum + cal_Quiescent_Conditions_Cost_Bias
ELSE
        ** Otherwise, there is no additional penalty
        Pcost_total_this_loop = Pcost_total_minimum
ENDIF
```

10. Recall from memory the independent variable with least cost from the last control loop.

The optimal input torque from the last loop, Ti_min_cost_last_loop, and its cost value, Pcost_total_last_loop are recalled from memory.

11. Compute the dependent variable at this operating point

Compute Pbat at Ti_min_cost_last_loop exactly as done in step 3. The result is Pbat_at_Ti_min_cost_last_loop.

12. Determine whether the dependent variable is within its constraints. If outside, skip to step 14.

Compare Pbat_at_Ti_min_cost_last_loop to Pbat_min and Pbat_max just as in step 4 and take the proper action.

13. Compute the Cost Function Value at this Operating Point

Just as in step 5, but compute the cost function at Ti_min_cost_last_loop.

14. Eliminate the Independent Variable with higher cost (or infeasible dependent variable)
(Eliminate the Input Torque, Ti, with higher cost (or infeasible Pbat))

Simply keep the input torque, Ti, with the lower cost between Ti_min_cost from step 8 Ti_min_cost_last_loop. Assign the value to Ti_min_cost_this_loop.

15. Store in memory the selected independent variable
(Store in memory, Ti_min_cost_this_loop.)